Peruvian sign language recognition using a hybrid deep neural network

Yuri Vladimir Huallpa Vargas[1][0000−0003−4108−2631], Naysha Naydu Diaz Ccasa[2][0000−0003−1202−474X], and Lauro Enciso Rodas[3][0000−0001−6266−0838]

Universidad Nacional de San Antonio Abad del Cusco, Av. de La Cultura 773, Peru
http://in.unsaac.edu.pe/home/

Abstract. Hearing people have the ability to communicate with their hands and interpret sign language (SL), but this builds a communication gap with normal people. There are models for SL recognition that have images sequence RGB as input; however, the movements of the body in 3D space is necessary to consider due to the complexity of the gestures. We built a model for Peruvian sign language recognition (PSL) to Spanish composed by 4 phases; first, the preprocessing phase in charge to process RGB, depth and skeleton streams obtained through the Kinect sensor v.1; second, the feature extraction which learn spatial information through 3 types of convolutional neural network (CNN); third, the bidirectional long short term memory (BLSTM) with residual connections in charge to reduced and encode the information. Finally, a decoder with attention mechanism and maxout network which learn the temporal information. Our proposed model is evaluated in LSA64 and ourself-built dataset. The experimental results show significant improvement compared to other models evaluated in these dataset.

Keywords: Sign language recognition · recurrent neural network · long short term memory · convolution neural network.

1 Introduction

Currently, people don't have the need to learn SL that is unique to each community and designed by themselves, i.e. the spanish SL is not completely similar as peruvian although them speak spanish. There are models that use CNN, Recurrent Neural Network (RNN) and Bidirectional Recurrent Neural Network (BRNN) that get good results for their SLs; however, they lose useful information by capturing only one type of data. Create a specific dataset for a SL is complicated because they have their own large dictionary; therefore, it is not possible to apply the same models and dataset for peruvian context. For developing an architecture that achieves an acceptable success rate, it is necessary to identify techniques as DNN (Deep Neural Network) and Natural Language Processing (NLP), for build our model with CNN along with RNN we used techniques as early stoping (ES), transfer learning (TL) and dropout; in addition to, our dictionary is done with expressions rather than just using simple words, to have better accuracy we use three types of stream: rgb, detph and skeleton which provide more signal information regarding distance and position of the gesture.

2 Related works

In the last decades a lot of research focused on the actions recognition that emulate the human capacity to interpret the outside world, they have generated different contributions for translating SL based on images. In [22] applies 3D convolution network in a sequence of videos in order to maintain temporal information unlike 2D CNN that losing temporal information. Another research [13] makes use RGB and 3D skeleton stream as input, uses VGG-16 CNN to extract features for each frame from RGB then both data are passed to different encoder-decoder with LSTM, their outputs are merged using a combination probability to improve accuracy. In [4] perform activity recognition and image tagging with VGG then it feed to LSTM to learn spatial-temporal information, the training and testing were performed at UCF101 dataset and verified that a recurrent neural network (RNN) with two layers achieves better results than single layer. [19] use a Time of Flight (TOF) sensor, the least relevant depth data were removed using principal component analysis (PCA) and persistent feature histogram [18] was used to feed a LSTM with softmax. In [16] propose an attention model with a LReLU activation function that only depends on the output hidden state of the LSTM, the LSTM have TanH and LRelU activation functions which accelerates the convergence; the disadvantage of this model is unable to distinguish the order of a sequence. [12] proposes two models of attention; global and local attention, which differ in data dependency of the input, global attention considers all hidden states of the encoder as long as local attention processes the hidden states through slide window while the inputs are processed. Finally, in [14] propose two approaches to infer gestures from dataset LSA64 [17]; first approach called Prediction Approach represents each video by a sequence of predictions made by an inception network and then passed to an RNN and the second model called Pool layer represents each video by a feature vector of size 2048 made by the last pooling layer and then it passed to the RNN.

3 Propose method

In this section we describe the model. First, we preprocess each frame from depth and skeleton stream since they have different distribution respect to necessary input for ResNet50 [8]; however, each frame from RGB stream is only resized. Second, we have 3 CNN (rgbResNet50, depthResNet50 and skeleton-ResNet50) which are fed by each frames from streams had already preprocessed, then extract the main features and joined them. Third, the joined features feed to encoder for reducing the dimensionality. Fourth, all output sequence from encoder feed to attention mechanism from decoder and for each time step t the attention pass context ctx towards LSTM from decoder to produce one output $\hat{y_t}$. At the end $\hat{y_t}$ is analized to interpret what SL is it, see Figure 1.

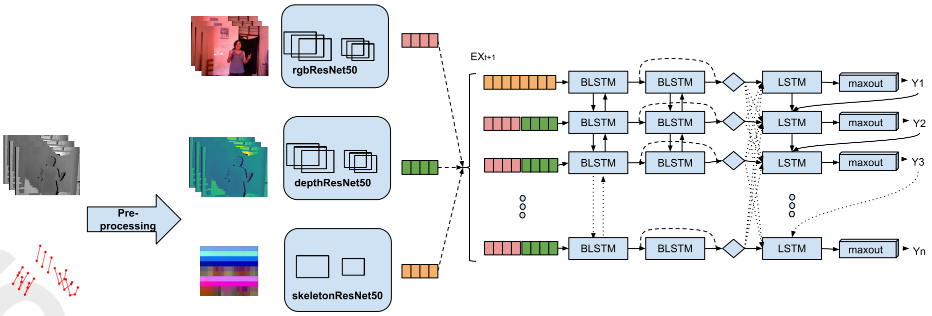

Fig. 1. Flow of the proposed method for Peruvian SL Recognition.

3.1 Preprocessing

We perform the preprocessing for our 3 data types and them are resized by area interpolation. First, RGB streams are only resized to 244 × 244 × 3 so that have similar distribution as needed input to ResNet50. To depth stream, let $D = (a_{ij})_{480X640}, \forall\, a_{ij} \in \mathbb{R}^2$ a matrix that represents a depth frame, for each position (a_{ij}) kinect return k, $\forall k \in \mathbb{N}$; k is transformed into a binary number of 16 bits with 3-bits shifted to the right to obtain the distance in millimeters and it is scaled to values between 0 to 1 by the Equation (1). Where D' is the scaled matrix, $\min(D)$ and $\max(D)$ is the minimum and maximum value respectively; then viridis [15] is used to map each position (a_{ij}) from D' in one RGB color, where purple indicates the closest pixel to the kinect sensor and yellow is the farthest. Finally, the result of size 480 × 640 × 3 is resized to 244 × 244 × 3.

$$D'(a_{ij}) = \frac{D(a_{ij})-\min(D)}{\max(D)-\min(D)} \qquad (1)$$

Each frame from skeleton stream consists of 20 (x, y, z) coordinates, we only consider 10 coordinates for upper body. To preprocess the skeleton stream a methodology similar to [11] is followed where the (x, y, z) coordinates are mapped into X, Y and Z matrix, before to stack $[X; Y; Z]$ to get one image $S = (a_{ijk})_{10\times F\times 3}$ where F is the number of frames in one stream, we apply the Equation (1); After, let $I_r = \lceil \frac{244}{10} \rceil$, $I_c = \lceil \frac{244}{F} \rceil$ where I_r and $I_c \in \mathbb{N}$, each row and column of S are repeated I_r and I_c times respectively to get a new $S = (a_{ijk})_{(I_r\times 10)\times(I_c\times F)\times(3)}$; at the end S is resized to 244 × 244 × 3.

3.2 Features extraction

The CNN are used in many researches [3, 10, 24] that show a satisfactory efficiency in the classification. We used pretrained ResNet50 as a basis to build our components (rgbResNet50, depthResNet50 and skeletonResNet50), removed the last fully conected layer (FC) of 1000 neurons, for depthResNet50 and skeletonResNet50 we added dropout with a keep-prop of 0.8, FC layer with 14 and 21 neurons to classify depth and skeleton images in our dataset; and them are retrained by applying transfer learning [1]. After training, the last average pooling layer of $7 \times 7 \times 2048$ are taken from the aforementioned components and applied max pooling with a kernel of 7×7 which returns a vector of 1×2048 considered as features vectors. Let $R_i = \nu(rgb_i)$, $D_i = \rho(depth_i)$ and $S_1 = \delta(skeleton_1)$ where ν, ρ y δ are the components, R_i and D_i are the i-th feature vector from RGB and depth stream, where $i = 1, \cdots, F$ and S_1 is the skeleton's feature vector. The feature vectors are combined by the Equation 2.

$$X_{F+1 \times 4096} = Concat \begin{pmatrix} S_1 & S_1 \\ R_1 & D_1 \\ \vdots & \vdots \\ R_F & D_F \end{pmatrix}_{F+1 \times 4096} \tag{2}$$

3.3 Encoder multilayer bidirectional LSTM with residual connections

LSTM [9] is a variation of RNN useful for dealing the problem of vanishing gradient, Equation 3. LSTM use the past context h_{t-1} to produce the output h_t, but for SL is necessary to know past and future context; hence, we used BRNN that can perform the aforementioned process by exploring the past and future context, its output is $h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t]$. [8] shows that adding skip connections between adjacent layers can accelerate training and achieve better results, see Fig. 2. Where x_t^i is input, h_t^i is hidden state, H_i is the BLSTM function associated with the i-th layer ($i = 1, \cdots, L$), L is number of layers and x_t^{i+1} is the element-wise added between x_t^i and h_t^i. According [3,7,21] the RNN with multiple layers stacking tends to achieve higher performance than shallower RNNs. Our encoder consists of 2 BLSTM and the last layer is a residual layer.

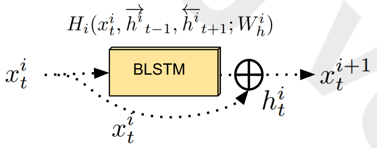
$H_i(x_t^i, \overrightarrow{h^i}_{t-1}, \overleftarrow{h^i}_{t+1}; W_h^i)$
x_t^i
BLSTM
x_t^{i+1}
h_t^i
x_t^i

Fig. 2. Bidirectional LSTM with residual connection.

$$\begin{gathered} i_t = \Theta\left(x_t W_i + h_{t-1} U_i + b_i\right) \\ f_t = \Theta\left(x_t W_f + h_{t-1} U_f + b_f\right) \\ o_t = \Theta\left(x_t W_o + h_{t-1} U_o + b_o\right) \\ \hat{c}_t = tanh\left(x_t W_c + h_{t-1} U_c + b_c\right) \\ c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t \\ h_t = o_t \odot c_t \end{gathered} \tag{3}$$

3.4 Attention decoder with maxout network

Attention mechanisms has been used in many researches [12, 16] since it generates a behavior similar to how a human interprets the outside world, focusing only on the relevant information. So our decoder incorporate attention mechanism [2] that return a context vector ctx_t witch depend on sequence output of encoder h_t and previous hidden state of decoder s_{t-1}, for each time step encoder return hidden state $s_t = LSTM(\hat{y}_{t-1}, s_{t-1}, [ctx_t, s_{t-1}])$, then each s_t pass to feed-forward maxout network [6] defined by the following equation:

$$\begin{aligned} \alpha_t(s) &= \max_{j\in[1,k]} z_{tj} \\ z_{tj} &= s^T W_{\dots tj} + b_{tj} \end{aligned} \tag{4}$$

$W \in \mathbb{R}^{d\times m\times k}$ y $b \in \mathbb{R}^{m\times k}$ are trainable weight, k is the number of linear neuron that compose a one hidden unit of maxout. Maxout network is quite robust to handle the problem of vanishing gradient however it is prone to overfitting; but dropout can be applied to control this problem and get better performance [20], we infer $\hat{y}_t = softmax(\alpha_t)$ by applying softmax clasifier on the output of maxout.

4 Experiments

In this section we perform a series of tests on the proposed model in ourself-built dataset VideoLSP10 [23] and LSA64 [17]; dataset was divided through hold-out. We trained the model in train set and evaluated the performance with four evaluation metrics in validation set, VideoLSP10 is made up of 3 dataset.

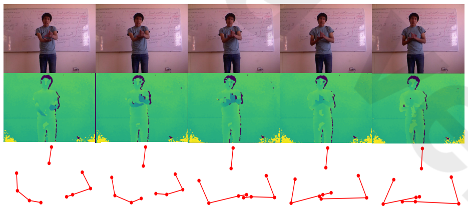

Fig. 3. RGB, Depth and Skeleton stream from LSP10.

First, depthLSP contains 2045 depth data divided into 14 classes to train the component depthResNet50 and is divided in the following way: 96% for training and 4% for validation. Second, skeletonLSP contains 1701 skeleton movements which was divided into 21 classes, each frame of a movement is composed by 10 coordinates (x, y, z), we used it for training the component skeletonResNet50 and is divided in the following way: 90% for training and 10% for validation. Finally, LSP10 contains rgb, depth and skeleton data, see Fig. 3; it consists of 600 videos divided in 10 classes of peruvian phrases and is divided in the following way: 84% for training and 16% for validation.

4.1 Hyperparameters

We use stochastic gradient descent (SGD) with a learning rate $\alpha = 0.001$ and momentum $\beta = 0.9$ for retraining depthResNet50 and skeletonResNet50.

Our proposed RNN have 2 BLSTM layers with 500 units in the encoder, the last layer contains residual connections; the decoder have attention mechanism with 64 units, one layer LSTM with 900 units, maxout network with 90 units where each linear units have 5 neurons and the output is softmax classifier. We applied dropout and recurrent dropout $keepProp = 0.5$ in encoder, dropout with $keepProp = 0.7$ in maxout network. Finally, the parameters are optimized using Root Mean Square Propagation (RMSprop) with a learning rate $\alpha = 0.045$, $\rho = 0.94$, $\epsilon = 1$ and learning decay= 0.00. Our rgbResNet50 linked to RNN is compared with the approaches [14], the parameters are optimized using Adam optimization with a learning rate $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and learning decay= 0.1. For each training are used mini-bach of 4 streams using the categorical-cross-entropy loss function and to prevent overfitting early stopping is used.

5 Results

Based on the experimentation, we obtain the results of the components and models evaluated in the validation set, see Table 1. DepthResNet50 trained in depthLSP dataset achieve 97.67 % of accuracy, for skeletonhResNet50 trained in skeletonLSP achieve 95.24% of accurancy. However, rgbResNet50 has no results since ResNet50 is used as FE; subsequently, the FE components were added to RNN (encoder linked to decoder). The result of rgbResNet50 linked to RNN in LSP10_{rgb} achieved 88.6% of accuracy, depthResNet50 linked to RNN in LSP10_{depth} achieved 85.3%. Finally, joining the 3 FEs to the RNN using the LSP10 data set achieved 99.2 % accuracy.

			%			
Components		**Dataset**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
FE	rgbResNet50	-	-	-	-	-
	depthResNet50	depthLSP	97.67	1	94.8	97.1
	skeletonResNet50	skeletonLSP	95.24	98.9	92.7	94.8
Models						
rgbResNet50+RNN		LSP10_{rgb}	88.6	88.8	84.0	85.8
depthResNet50+RNN		LSP10_{depth}	85.3	84.2	81.4	81.7
FE+RNN		LSP10	**99.2**	**99.0**	**99.5**	**99.2**

Table 1. Results in the validation set for FE components and models, using the VideoLSP10 dataset.

Our model rgbResNet50 linked with RNN has been compared with proposed models in [14]: Prediction Approach and Pool Layer Approach using the LSA64 dataset, Our model achieved **98.44** % of accuracy, Pool Layer Aproach achieved 95.21 % of accuracy and Prediction Approach achieved 80.87 % of accuracy.

6 Conclusions

We propose a hybrid model based on CNN and RNN which incorporates advanced models such as maxout, attention mechanisms and residual connections. The model is composed of four parts. Firstly, the preprocessing phase of RGB, depth and skeleton. Secondly, feature extraction phase that use CNN based in ResNet50. Thirdly, encoder phase that reduce the inputs dimensionality. Finally, a decoder to relate the temporal information and infer what sign it is. The model is able to translate a phrase from the VideoLSP10 dataset achieving 99.2% of accuracy and in LSA our model achieve an accuracy of 98.44% only using RGD data. Hence, our model can be used for any sign language since in both dataset an acceptable accuracy are achieved and also VideoLSP10 is challenger than LSA64. Eventually future researches may improve the model and apply it to a larger dataset.

References

1. Akilan, T., Wu, Q.M.J., Yang, Y., Safaei, A.: Fusion of transfer learning features and its application in image classification. In: 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE). pp. 1–5 (April 2017). https://doi.org/10.1109/CCECE.2017.7946733
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv e-prints **abs/1409.0473** (Sep 2014), https://arxiv.org/abs/1409.0473
3. Cai, M., Liu, J.: Maxout neurons for deep convolutional and lstm neural networks in speech recognition. Speech Communication **77** (12 2015). https://doi.org/10.1016/j.specom.2015.12.003
4. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. CoRR **abs/1411.4389** (2014), http://arxiv.org/abs/1411.4389
5. Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 1019–1027. Curran Associates, Inc. (2016), http://papers.nips.cc/paper/6241-a-theoretically-grounded-application-of-dropout-in-recurrent-neural-networks.pdf
6. Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. In: Dasgupta, S., McAllester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning. pp. 1319–1327. No. 3 in Proceedings of Machine Learning Research, PMLR, Atlanta, Georgia, USA (17–19 Jun 2013), http://proceedings.mlr.press/v28/goodfellow13.html
7. Graves, A., Mohamed, A., Hinton, G.E.: Speech recognition with deep recurrent neural networks. CoRR **abs/1303.5778** (2013), http://arxiv.org/abs/1303.5778

8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**, 1735–80 (12 1997). https://doi.org/10.1162/neco.1997.9.8.1735
10. Huang, S., Mao, C., Tao, J., Ye, Z.: A novel chinese sign language recognition method based on keyframe-centered clips. IEEE Signal Processing Letters **25**(3), 442–446 (March 2018). https://doi.org/10.1109/LSP.2018.2797228
11. Laraba, S., Brahimi, M., Tilmanne, J., Dutoit, T.: 3d skeleton-based action recognition by representing motion capture sequences as 2d-rgb images. Computer Animation and Virtual Worlds **28** (05 2017). https://doi.org/10.1002/cav.1782
12. Luong, M., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. CoRR **abs/1508.04025** (2015), http://arxiv.org/abs/1508.04025
13. Mao, C., Huang, S., Li, X., Ye, Z.: Chinese sign language recognition with sequence to sequence learning. In: Yang, J., Hu, Q., Cheng, M.M., Wang, L., Liu, Q., Bai, X., Meng, D. (eds.) Computer Vision. pp. 180–191. Springer Singapore, Singapore (2017)
14. Masood, S., Srivastava, A., Thuwal, H.C., Ahmad, M.: Real-time sign language gesture (word) recognition from video sequences using cnn and rnn. In: Bhateja, V., Coello Coello, C.A., Satapathy, S.C., Pattnaik, P.K. (eds.) Intelligent Engineering Informatics. pp. 623–632. Springer Singapore, Singapore (2018)
15. Nathaniel Smith, S.v.d.W.: mpl colormaps. https://bids.github.io/colormap/ (2015)
16. Raffel, C., Ellis, D.P.W.: Feed-forward networks with attention can solve some long-term memory problems. CoRR **abs/1512.08756** (2015), http://arxiv.org/abs/1512.08756
17. Ronchetti, F., Quiroga, F., Estrebou, C., Lanzarini, L., Rosete, A.: Lsa64: A dataset of argentinian sign language. XX II Congreso Argentino de Ciencias de la Computacin (CACIC) (2016)
18. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3384–3391 (Sept 2008). https://doi.org/10.1109/IROS.2008.4650967
19. Sarkar, A., Gepperth, A., Handmann, U., Kopinski, T.: Dynamic hand gesture recognition for mobile systems using deep lstm. In: Horain, P., Achard, C., Mallem, M. (eds.) Intelligent Human Computer Interaction. pp. 19–31. Springer International Publishing, Cham (2017)
20. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**, 1929–1958 (2014), http://jmlr.org/papers/v15/srivastava14a.html
21. Su, P., Ding, X., Zhang, Y., Miao, F., Zhao, N.: Learning to predict blood pressure with deep bidirectional LSTM network. CoRR **abs/1705.04524** (2017), http://arxiv.org/abs/1705.04524
22. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: C3D: generic features for video analysis. CoRR **abs/1412.0767** (2014), http://arxiv.org/abs/1412.0767
23. Vargas, Y.V.H.: Peruvian sign language videolsp10. https://github.com/videoLSP/VideoLSP10 (2019)
24. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. CoRR **abs/1502.03044** (2015), http://arxiv.org/abs/1502.03044